Research Paper | Computer Science | E-ISSN No : 2454-9916 | Volume : 3 | Issue : 5 | May 2017

PROPOSED METHOD FOR WORKLOAD BALANCING USING GRID COMPUTING

Kirti Rathi [1] | Dr. Kanwal Garg [2]

[1] Research Scholar, M.Tech. (CSE), Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, Haryana, India.

[2] Supervisor, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, Haryana, India.

ABSTRACT

The premise of this paper is to specify the Workload Balancing in grid computing environment. Grid computing has become a popular paradigm for next generation parallel and distributed computing. In this paper the researcher proposes a load balancing and outlines a new method to resolve workload balancing problem in grid computing. The main objective of this paper is to reduce overhead and achieve high performance with the help of scheduling in grid environment.

KEYWORDS: Grid Computing, Load Balancing, Simulation, Nimrod-G.

1. INTRODUCTION

Grid Computing

Grid Computing is all about many distributed processors in a global manner and shares various computational resources to solve various problems [4]. There are various issues related with the Grid computing is coordination and sharing of resources as well as Dynamic nature of resources and virtual organizations of resources. A computational Grid is a software and hardware infrastructure that provides consistent, dependable and reliable inexpensive access to high end computational capacity [3]. Grid Computing is concerned with the sharing of coordinated resources and solution of problem due to dynamic nature in multi-institutional virtual organization. Virtual organization is concerned with the set of individuals or institutions defined by some sharing rules. Grid Computing is satisfied with the following features:

- Coordination of several resources that are not related to the centralized control.
- Using open, standard and general purpose protocols and interfaces.

Grid architecture consists of following layers:

- Fabric Layer: This provides local information or services of resources.
- Connective Layer: It provides core communication and authentication of protocols.
- Resource Layer: It enables sharing of resources.
- Collective Layer: It provides coordination, interactions across various resources.
- Application Layer: Application Layer is used by user applications that use collective, resource and connective layer to perform grid operations in virtual organizations [4].

2. LOAD BALANCING TYPES

In a distributed system will have a number of interconnected resources who work independently or in the cooperative nature. In distributed system each resource has its own workload owner that represents an amount of work that has to be performed and everyone may have different capability of processing. For the minimization of the time needed to perform any task, the workload has to be equally distributed among all the resources based on their speed. The main objective of Load Balancing acquire the average response time of applications which means maintaining the workload proportionally [1].

Load Balancing algorithms can be classified into two categories:

- **Static Load Balancing**

Static Load Balancing allocate task to resources at the compile time based on no prior task information. In Static Load balancing task is assigned to a resource when it is generated or committed to the system using a fixed schema. In Static Load balancing computational work of each processor must be balanced so that no processor will have to wait for others to compute. Tasks are assigned to processors during the compile time and their relation is determined [5].

- **Dynamic Load Balancing**

Dynamic Load Balancing is based on application level load balancing for the individual parallel jobs. It confirms that all the loads in the Dynamic Load Balancing are distributed in such a method that overall load in the system get balanced and application programs reach maximum benefit from the available resources [2].

Dynamic Load Balancing has two major versions:

- System agent: It collects information related to the system such as load of the total system as well as communication latency between the processors.
- Dynamic Load Balancing agent: Dynamic Load Balancing agent is used to perform the load balancing

3. LOAD BALANCING POLICIES

Load Balancing algorithms can be defined by various policies for their implementation:

1. Information policy: It specifies what load information to be collected, when is to be collected and from where is to be.
2. Triggering Policy: It specifies the appropriate timing to start a Load Balancing operation.
3. Resource Policy: It classifies a resource as a server or tasks receiver according to its status availability and capabilities.
4. Location Policy: It uses output of the resource type policy to find a suitable location for a server or receiver.
5. Selection Policy: It defines tasks that should be transferred from overloaded resources to unloaded resources [6].

4. LOAD BALANCING APPROACHES

1. Local versus Global

In Local Load Balancing the scheduling concept determines how the processes or task occurs on a single CPU and execution of processes. Local Load Balancing uses local information's to decide upon a Load transfer or migration. On the other hand Global load balancing workload information is transfer between the system elements. Scheduling discipline method uses load information about the system to allocate task or processes to multiple processors to obtain a system-wide performance [7].

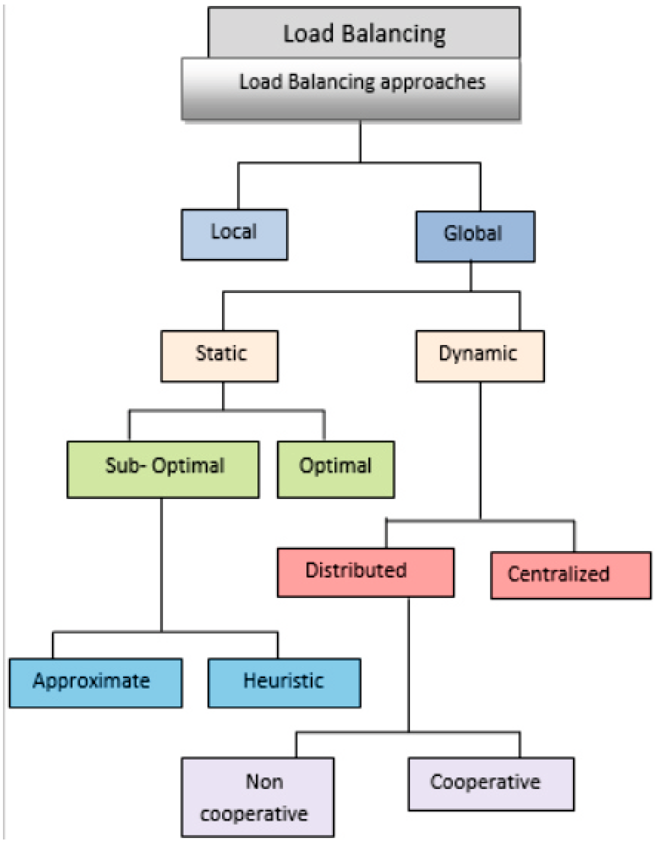

Figure1: Load Balancing approaches

2. Optimal versus Sub-optimal

All information related to the state of resources and allocation of jobs is known. An optimal assignment has to be made based on some defined criterion function such as maximum throughput, minimum makespan and maximum resource utilization. Scheduling algorithms has NP- complete nature and suffers from the difficulty of reasonable assumptions which are required to prove the optimality of an algorithm in Grid scenarios. Suboptimal solutions can further be divided into two categories:

3. Appropriate versus Heuristic

In the Dynamic Load Balancing approach the appropriate algorithm used in formal computational models. This approach satisfied when a sufficiently good solution is found rather than searching for the entire solution space for an optimal solution. In this case where a metric is available for evaluating a solution this technique is used to reduce the time taken for an appropriate solution.

4. Centralized versus Distributed

In the Dynamic Load Balancing the decision for making global reduction may lie with one centralized location or can be shared with multiple distributed locations. In dynamic load balancing centralized has many advantages of implementation but there is lack of scalability, fault tolerance and becoming a bottleneck of performance. In the distributed approach of load balancing the state of each resource is distributed among the various nodes that are responsible for managing their own resources or allocation of tasks within their queues to the other nodes [8].

5. Cooperative versus Non-Cooperative

This approach is based on the technique that if the adoption of Load Balancing is possible then working of nodes independently or correctly involved in job balancing will be the next issue. In non- cooperative an individual system makes their own decisions regarding their own objectives independently and acts as alone autonomous entities.

5. Proposed System Architecture

The author proposed a technique on the basis of Load Balancing under a Grid Computing environment which balances Load dynamically occurs due to the dynamic nature. Load imbalance occurs due to the dynamic nature or some changes occur in the Load state [9].

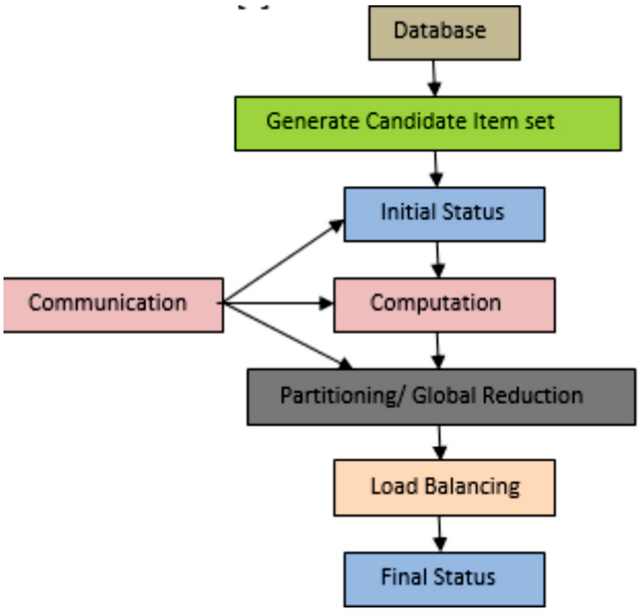

Figure 2: Proposed Architecture

In the above architecture the whole data is stored in the database through which we can find or generate item sets. Now check the initial status that collect the whole information regarding all the connected nodes like items entry. After this computation has to be done that is used for sending and receiving messages. Now partitioning is to be performed that reduces or partitions the load globally [10]. Finally Load Balancing or can say Dynamic Load Balancing is done based on the patterns involved in the item set. A threshold value is defined for the Load estimation that is upper bound and lower bound. Upper bound means load should not exceed more than two and lower bound means load should not exceed than one. Now calculation of total time spent on the movement of data, communication between the item set and Load Balancing that can be viewed as overhead due to parallel processing is done. Finally check the status [11].

4. Proposed Methodology

In this research work the researcher verifies the workload issues and works on the performance domain. Dynamic nature imbalances workload the researcher proposed a method which is the combination of the Simulation model and Nimrod-G.

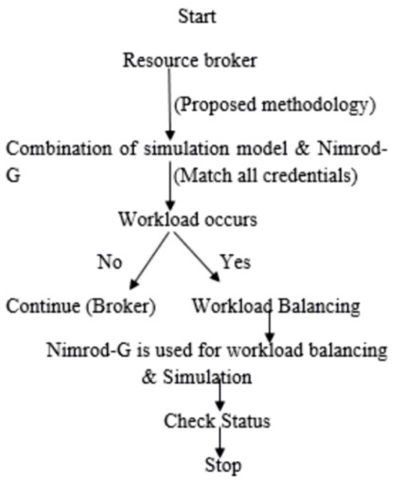

Figure3: Flowchart of proposed methodology

Initially process starts at the beginning with the proposed methodology. The proposed methodology consists of resource broker. Firstly user interacts with the resource broker which consists of complexities of Grid Computing environment. Further resource broker having users interacts with the resources having negotiation for cost using services, mapping of tasks to the resources, allocates stages to the application and finally starts allocation of jobs and execution.

In this paper researcher proposed a method that is the combination of Simulation model and Nimrod-G. Simulation model supports heterogeneous Grid resources (both time and space shared) users and application models. Nimrod-G handles the tasks performed by the resource broker and evaluation of performance and various cost and time minimization scheduling algorithms.

Load Balancing take place when some changes occur in the state of load. As users interacts with the resources with the various complexities. Tasks are assigned to the resources. Sometime new resource arrived before the completion of any task and withdrawal of existing resource may occur at same time. In this method if workload does not occur then process continues to the resource broker and interaction between the user and the resources continues. On the other hand if workload occurs then there is needed to balance the workload. Then simulation model and Nimrod-G is used for the modeling and scheduling process completion of tasks takes place in appropriate manner. Nimrod-G balances the execution of resources with the users having tasks and evaluates performance and budget constraint in cost, time and space, manner. After this whole process, finally status has to be checked so that it balances the workload along with the time, space and execution of jobs or tasks in appropriate manner. Nimrod-G performs tasks with the Simulation method along with the different models so that allocation of jobs and tasks to the resources should be in proper manner.

Simulation model provides simulation and modeling of resources and connection of networks with the different capabilities, configurations, queues, and domains. This model provides information services which contain properties of job program, processing requirements, the size of input and output files and required interface for mapping to the resources. Simulation model is used for the management of resources for performing execution of any task.

7. CONCLUSION

Grid Computing has become a popular paradigm for next generation parallel and distributed computing. It is impossible to perform a scheduler performance in controllable and repeatable manner as users and resources are distributed across multiple domains or organizations [12]. Then the researcher presented Load Balancing types and policies. Also the researcher presented Load Balancing approaches. In this paper the researcher contributes to the overall body of research concerning performance in Grid Computing and provides an overview of Workload Balancing. The researcher further proposes a method to resolve the Workload balancing method by using Simulation model and Nimrod-G. Further various performance procedures must be developed in future for better results in Grid Computing environment.

REFERENCES

[1] Anuradha Sharma, Seema Verma," Survey Report on Load Balancing in Grid Computing Enviornment", International Journal of Advanced research & Studies, Vol 4, No. 2, Jan-March, 2015.

[2] Belabbas Yagoubi and Yahya Slimani," Dynamic Load Balancing Strategy for Grid Computing," World Academy of science ,engineering and Technology 19, 2006.

[3] Frederic Magoules, Thi-Mai-Houng Nguyen and Lei Yu, Grid Resource Management Toward Virtual and Services Complaint Grid Computing, CRC Press Taylor & Francis Group (2009).

[4] Frederic Magoules, Jie pan, Kiat-An Tan and Abhinit Kumar, Introduction to Grid Computing, CRC Press Taylor & Francis Group, (2009).

[5] K. Devine, E. Boman, R. Heaphy and B. Hendrickson," New Challenges in Dynamic Load Balancing". Appl. Num. Maths, Vol.52, issues 2-3, 133-152. (2005).

[6] Kabalan, K.y, W.W, Smar and J.Y Hakimian, 2002, Adaptive Load Sharing in heterogeneous.

[7] Leinbeuger, W., G. Karypis, V. Kumar and R. Biswas, 2000, Load Balancing across near homogeneous multi-resource servers. In 9th heterogeneous Computing Workshop, pp: 60-71.

[8] T.G Casavant and J.L Khul," Taxonomy of Scheduling in general purpose distributed computing systems," IEEE transactions on software engineering, 14(2): pp: 140-155, 1995.

[9] Pawandeep Kaur, Harshpreet singh," Performance Analysis of Adaptive Dynamic Load Balancing in Grid Computing environment," International Journal of computer science and information technologies Vol. 3(3), 2012 , 4473-4479.

[10] Raja Tlili and Yahya Slimani, Dynamic Load balancing for large-scale distributed association rule mining, In proceedings of the 9th International Symposium on performance Science (Algieris, Algeria), (2009) May.

[11] M. H. Willebeek-LeMair and A. P. Reeves, "Strategies for Dynamic Load Balancing on Highly Parallel Computers,". IEEE Transactions on Parallel and Distributed Systems, Vol. 4, No. 9, pages 979-993, September, (1993).

[12] I Foster and C. Kesselman, The Grid2: Blue print for a New Computing Infrastructure, Morgan Kaufmann, (2003)